

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-016392

(43)Date of publication of application : 17.01.1997

(51)Int.CI.

G06F 9/06

G06F 9/06

G06F 17/30

(21)Application number : 07-161168

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 27.06.1995

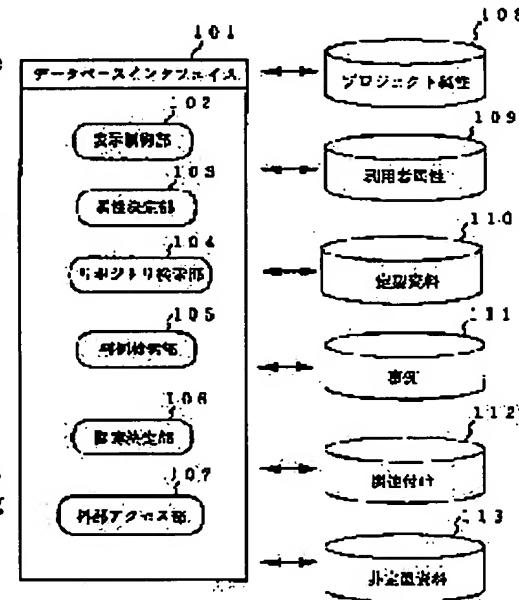
(72)Inventor : HOSONO YOSHIHISA

(54) SOFTWARE DEVELOPMENT SUPPORT SYSTEM

(57)Abstract:

PURPOSE: To provide the information of the related result in accordance with corrected contents at the time of correcting an arbitrary result by providing an integrated database for storing relating information for mutually relating all the results obtained in respective development phases.

CONSTITUTION: A typical source material database 110 stores all the results in the respective development phases of system development to be the objects of template preparation such as system constitution, function design, source code, object code, execution format, unit test specification, unit test result, combination test specification, combination test result, overall test specification and overall test result. Besides, a relating database 112 stores the information of relating between all the data stored in the routine material database 110 and a non routine material database 113. Then, all the results provided in the development phases are related so that the related module can be detected and presented in the case of changing a certain module of a software to be developed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-16392

(43)公開日 平成9年(1997)1月17日

(51)Int.Cl. ⁶ G 06 F 9/06 17/30	識別記号 5 4 0 5 3 0	府内整理番号 9289-5L	F I G 06 F 9/06 15/413	技術表示箇所 5 4 0 U 5 3 0 T 3 1 0 A
--	------------------------	-------------------	------------------------------	---

審査請求 未請求 請求項の数12 OL (全 19 頁)

(21)出願番号 特願平7-161168

(22)出願日 平成7年(1995)6月27日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 細野 善久

尼崎市塚口本町八丁目1番1号 三菱電機
株式会社産業システム研究所内

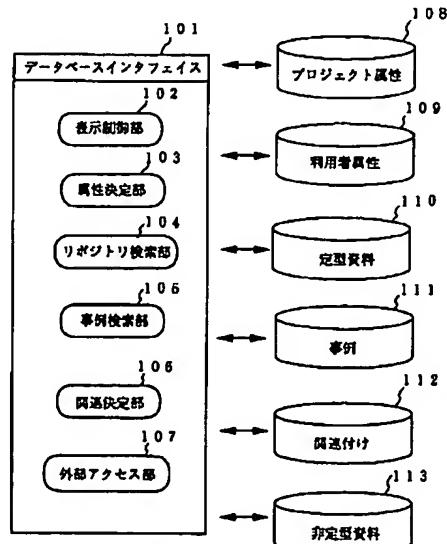
(74)代理人 弁理士 田澤 博昭 (外2名)

(54)【発明の名称】 ソフトウェア開発支援方式

(57)【要約】

【目的】 任意の成果物を修正する際に、その修正内容に応じて関連する成果物の情報を提示するソフトウェア開発支援方式を得ることを目的とする。

【構成】 要求分析、システム設計、機能設計、詳細設計、ソースコード生成生成、オブジェクトコード生成、システム結合、単体試験、結合試験、総合試験、運用試験の各開発フェーズで得られる全ての成果物に関する情報を格納する定型資料データベース110と、成果物の相互の関連付け情報を格納する関連付けデータベース112とを設ける。



108 : プロジェクト属性データベース (統合データベース)

109 : 利用者属性データベース (統合データベース)

110 : 定型資料データベース (統合データベース)

111 : 事例データベース (統合データベース)

112 : 関連付けデータベース (統合データベース)

113 : 非定型資料データベース (統合データベース)

【特許請求の範囲】

【請求項1】 要求分析、システム設計、機能設計、詳細設計、ソースコード生成、オブジェクトコード生成、システム結合、単体試験、結合試験、総合試験、運用試験の各開発フェーズで得られる全ての成果物に関する情報と、前記成果物を相互に関連付ける関連付け情報と共に格納する統合データベースを設けたソフトウェア開発支援方式。

【請求項2】 少なくとも前記機能設計及び前記詳細設計の開発フェーズにおいて、既定値を含むテンプレートを用いて開発を行うテンプレート利用開発ステップを含むことを特徴とする請求項1記載のソフトウェア開発支援方式。

【請求項3】 打合せ議事録、打合せ資料、及び参考文献を少なくとも含む、各前記開発フェーズの直接の成果物以外の資料と、各前記開発フェーズにおいて生成される成果物とを相互に関連付ける関連付け情報を前記統合データベースにおいて管理する非定型資料関連付けステップを含むことを特徴とする請求項1記載のソフトウェア開発支援方式。

【請求項4】 ソフトウェアを少なくともサブシステム、モジュールに細分化して、その各々についての版とそれらを組み合わせた全体の版との管理を行い、且つ、前記関連付け情報の改訂履歴を含めた版管理を行う版管理ステップを含む請求項1から請求項3のうちのいずれか一項記載のソフトウェア開発支援方式。

【請求項5】 既定値を含む前記テンプレートを用いて、ソフトウェア開発の各開発フェーズ毎にソフトウェアの機能、性能の検証を行う検証ステップを含むことを特徴とする請求項2記載のソフトウェア開発支援方式。

【請求項6】 ソフトウェア開発の前記各開発フェーズ毎に、前記テンプレートの既定値が設計値で置き換わった割合によりシステムを構成する要件が全てそろっているか否かを判定する開発進捗判定ステップを含むことを特徴とする請求項2記載のソフトウェア開発支援方式。

【請求項7】 利用者の参加するプロジェクト、プロジェクト内のチーム、及び前記利用者が担当する部分に関する情報を少なくとも管理し、前記利用者が利用を開始するときに入力する情報に基づいて、前記統合データベースから前記利用者の担当する成果物を呼び出す利用者関連情報呼び出しステップを含むことを特徴とする請求項1記載のソフトウェア開発支援方式。

【請求項8】 開発中または開発済みのシステムの一部又は全部の機能を変更する際に、システムの成果物の間のシステム構成上の関係に基づいて、前記変更に直接関与する部分の改造が影響する範囲の見積る影響範囲見積ステップを含むことを特徴とする請求項1記載のソフトウェア開発支援方式。

【請求項9】 前記影響範囲の見積とソフトウェア開発の各利用者の担当部分の情報とから、前記改造の部分の

担当者を検索し提示する担当者検索ステップを含むことを特徴とする請求項8記載のソフトウェア開発支援方式。

【請求項10】 前記影響範囲の見積と各成果物の工数とにに基づき、前記改造の工数の見積を行う改造工数見積ステップを含むことを特徴とする請求項8記載のソフトウェア開発支援方式。

【請求項11】 利用者が入力する開発対象のシステムの特徴に基づいて、過去の開発事例から前記開発対象のシステムと関連の深い開発事例を選択し該開発事例の成果物をテンプレートとして使用する事例利用ステップを含むことを特徴とする請求項1記載のソフトウェア開発支援方式。

【請求項12】 過去の開発事例について前記非定型資料関連付けステップにおいて管理される前記開発事例の関連付け情報を参照することにより、前記開発事例に関連する情報を提示する開発事例設計根拠提示ステップを含むことを特徴とする請求項3記載のソフトウェア開発支援方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、計算機システムの構築の効率を向上し作業を容易にするためのソフトウェア開発支援方式に関するものである。

【0002】

【従来の技術】図27は計算機システムを開発するための従来のソフトウェア開発支援方式の代表的な例であるCASE(Computer Aided Software Environment)ツールにおけるソフトウェアプロセスモデルの一例であるVカーブモデルの開発の流れを示す図である。

【0003】次に動作について説明する。かかるソフトウェア開発支援方式では、ソフトウェアの開発は、要求分析(問題の解説)、システム(概要)設計、機能(基本)設計、詳細(モジュール)設計、ソースコード生成、オブジェクトコード生成、システム結合、単体試験、結合試験、総合試験、運用試験の各開発フェーズを順に進行する。各開発フェーズでは、各種の仕様書や、ソースプログラム、試験データ等が成果物として生成される。それらの成果物は、開発対象となる計算機システムにより異なる。また、ある開発フェーズで不具合が生じた場合には、すぐ上流の開発フェーズに戻るようにしてソフトウェア開発は進められる。

【0004】従来のソフトウェア開発支援方式を用いたCASEツールは上記した開発フェーズを支援するために、構造化手法、オブジェクト指向分析と言った方法論に基づいてソフトウェア開発の作法を規定し、またこれら方法論で採用されているドキュメントの記法を用いて仕様書を作成するためのエディタを提供している。例えば構造化手法を採用するものは、データフローダイアグラム(以下DFDと略記する)等を描画するためのエデ

イタを持ち、オブジェクト指向分析を採用するものはクラス階層図等のエディタを持っている等である。これらは単に描画のみではなく、描画された内容が各設計手法の枠組に添っているかの検査を行う機能を持つ。例えば、DFDではデータの流れが完結しているか等の検査、クラス階層図では継承関係がループしていないかどうか等の検査を行う等である。また、上記したように、これらの設計内容に応じて部分的又は完全なソースコードの生成を行う機能を有している。さらに、生成されたソースコードに対して異常動作が起きるかどうかの検査項目を生成する機能を有している。

【0005】さらに、従来のソフトウェア開発支援方式を用いたCASEツールは、上記した個々の開発フェーズ間の成果物の整合性のチェック機能と、連続した開発フェーズの間での成果物の変換のための機能を有している。整合性のチェックは各開発フェーズ毎に成果物の構成要素が全てそろった時点で可能になる。一般に、開発フェーズのうち、分析や設計、製作に関するもの間では、下流工程後のものほどより詳細かつ具体的な記述が行われる。CASEツールの中には、作成中の計算機システムに要求される機能、性能が満たされているかの評価機能を備えているものがあるが、これについても全ての構成要素に関する詳細な記述が完了している必要がある。特に、性能に関する評価は詳細設計が終了するまではできない。さらに、各開発フェーズで生成される成果物は個別システム毎に管理されており、連続して複数のシステムを作成するような場合でも、システム間での統合的な管理は行われることはない。

【0006】

【発明が解決しようとする課題】従来のソフトウェア開発支援方式は以上のように構成されているので、全ての成果物のモジュール等の設計の詳細化が完了してから機能、性能の評価を行い、機能が不十分な成果物のモジュールの修正に加えてその影響を吸収するために他の成果物のモジュール等の修正を行って全体の整合を取らなければならず、その関連ある成果物のモジュール等を探し出すために多大な手間がかかってしまうという問題点があった。

【0007】また、連続して複数のシステムを作成する際に以前に開発したシステムに類似の機能があったとしても、それを後から開発するシステムに流用して開発を効率的に行うことができないという問題点があった。

【0008】さらに、システム開発中又は保守にあたりシステムの機能を変更する際には、その機能を直接実現しているモジュールを改造するだけではなく、関連するモジュールについても前者のモジュールの変更内容の吸収するための改造が必要になる事があるが、このような場合、従来のソフトウェア開発支援方式では仕様変更の工数の見積りが困難であり、また、仕様変更に対する適切な判断材料を提示できないなどの問題点があった。

【0009】この発明は上記のような問題点を解消するためになされたもので、任意の成果物を修正する際に、その修正内容に応じて関連する成果物の修正を容易に達成するために、関連する成果物の情報を提示するソフトウェア開発支援方式を得ることを目的とする。

【0010】またこの発明は、類似したシステムの成果物等をテンプレートとして提供して開発効率を向上させ得るソフトウェア開発支援方式を得ることを目的とする。

【0011】さらにこの発明は、成果物の変更に伴って改造が必要な影響範囲を見積り、仕様変更に伴う改造の工数を見積ることにより、仕様変更に対する適切な判断材料を提供できるソフトウェア開発支援方式を得ることを目的とする。

【0012】

【課題を解決するための手段】請求項1の発明に係るソフトウェア開発支援方式は、各開発フェーズで得られる全ての成果物を相互に関連付ける関連付け情報を格納する統合データベースを設けたものである。

【0013】請求項2の発明に係るソフトウェア開発支援方式は、少なくとも機能設計及び詳細設計の開発フェーズにおいて、既定値を含むテンプレートを用いて開発を行うものである。

【0014】請求項3の発明に係るソフトウェア開発支援方式は、各開発フェーズの直接の成果物以外の資料と、各開発フェーズにおいて生成される成果物とを相互に関連付ける関連付け情報を統合データベースにおいて管理するものである。

【0015】請求項4の発明に係るソフトウェア開発支援方式は、サブシステム、モジュールの各々についての版とそれらを組み合わせた全体の版との管理を行い、且つ、関連付け情報の改訂履歴を含めた版管理を行うものである。

【0016】請求項5の発明に係るソフトウェア開発支援方式は、既定値を含むテンプレートを用いて、ソフトウェア開発の各開発フェーズ毎にソフトウェアの機能、性能の検証を行うものである。

【0017】請求項6の発明に係るソフトウェア開発支援方式は、ソフトウェア開発の各開発フェーズ毎に、テンプレートの既定値が設計値で置き換わった割合によりシステムを構成する要件が全てそろっているか否かを判定するものである。

【0018】請求項7の発明に係るソフトウェア開発支援方式は、利用者が利用を開始するときに入力する情報に基づいて、統合データベースから利用者の担当する成果物を呼び出すものである。

【0019】請求項8の発明に係るソフトウェア開発支援方式は、システムの成果物の間のシステム構成上の関係に基づいて、開発中または開発済みのシステムの一部又は全部の機能変更に直接関与する部分の改造が影響す

る範囲を見積るものである。

【0020】請求項9の発明に係るソフトウェア開発支援方式は、影響範囲の見積とソフトウェア開発の各利用者の担当部分の情報とから、改造の部分の担当者を検索し提示するものである。

【0021】請求項10の発明に係るソフトウェア開発支援方式は、影響範囲の見積と各成果物の工数とに基づき、改造の工数の見積を行うものである。

【0022】請求項11の発明に係るソフトウェア開発支援方式は、利用者が入力する開発対象のシステムの特徴に基づいて、過去の開発事例から開発対象のシステムと関連の深い開発事例を選択し該開発事例の成果物をテンプレートとして使用するものである。

【0023】請求項12の発明に係るソフトウェア開発支援方式は、過去の開発事例について非定型資料関連付けステップにおいて管理される開発事例の関連付け情報を参照することにより、開発事例に関連する情報を提示するものである。

【0024】

【作用】請求項1の発明におけるソフトウェア開発支援方式は、要求分析、システム設計、機能設計、詳細設計、ソースコード生成、オブジェクトコード生成、システム結合、単体試験、結合試験、総合試験、運用試験の各開発フェーズで得られる全ての成果物の間での関連づけを行うことで、開発するソフトウェアの例えはあるモジュールを変更する際に関連するモジュールを検出し提示することが可能になる。関連づけは利用者が関連づけられたモジュールを確認するために呼び出すためにも使用できる。

【0025】請求項2の発明におけるソフトウェア開発支援方式は、少なくとも機能設計及び詳細設計の開発フェーズにおいて、既定値を持つテンプレートを使用してソフトウェア開発を進める。これにより、システムの設計の際の指針を利用者に与えることができ、設計の効率を向上させ得る。また、テンプレートの利用は設計時の漏れも防ぐことができる。

【0026】請求項3の発明におけるソフトウェア開発支援方式は、打合せ議事録、打合せ資料、及び参考文献を少なくとも含む、各開発フェーズの直接の成果物以外の資料と、各開発フェーズにおいて生成される成果物とを相互に関連付ける関連付け情報を統合データベースにおいて管理する。このように、関連づけが打合せ議事録等にも及ぶことで、設計内容の背景の確認が容易になり、設計方針の誤った解釈によるミスを減少させることができる。

【0027】請求項4の発明におけるソフトウェア開発支援方式は、ソフトウェアを少なくともサブシステム、モジュールに細分化して、その各々についての版とそれらを組み合わせた全体の版との管理を行い、且つ、関連付け情報の改訂履歴を含めた版管理を行う。最小単位毎

に版管理を行うことで、関連付けの単位と版管理の単位が一致することになり、関連付けを版毎に管理することが可能になる。関連付け情報を実体と分離して管理することで、柔軟な関連付けの管理が行え、外部から取り込まれたデータに対しても関連付けが行える。

【0028】請求項5の発明におけるソフトウェア開発支援方式は、既定値を含むテンプレートを用いて、ソフトウェア開発の各開発フェーズ毎にソフトウェアの機能、性能の検証を行う。既定値の埋め込まれたテンプレートを使用することで、個別のモジュールの内容を検討するにあたって、そのモジュールについては新たに設計した内容を用いて、残りのモジュールについては既定値を用いて動作の模擬を行うことが可能になる。テンプレート上で機能、設計の見積を行うことにより、部分的な設計がシステム全体の要求に合致しているかを確認できる。

【0029】請求項6の発明におけるソフトウェア開発支援方式は、ソフトウェア開発の各開発フェーズ毎に、テンプレートの既定値が設計値で置き換わった割合によりシステムを構成する要件が全てそろっているか否かを判定する。より具体的には、各モジュールに付けられた設計が済であるか否かのマークを集計することで、システム開発の進捗を把握するための指標を提示することが可能になる。

【0030】請求項7の発明におけるソフトウェア開発支援方式は、利用者の参加するプロジェクト、プロジェクト内のチーム、及び利用者が担当する部分に関する情報を少なくとも管理し、利用者が利用を開始するときに入力する情報に基づいて、統合データベースから利用者の担当する成果物を呼び出す。従って、利用者がシステムの使用開始時に入力する情報をプロジェクト、利用者の属性と照合し、当該作業に必要な情報を検索、収集して提示することが可能になる。

【0031】請求項8の発明におけるソフトウェア開発支援方式は、開発中または開発済みのシステムの一部又は全部の機能を変更する際に、システムの成果物の間のシステム構成上の関係に基づいて、変更に直接関与する部分の改造が影響する範囲を見積る。従って、特定モジュールの変更のシステム全体への影響の度合いを見積り、仕様変更に対する評価基準を提示することができる。

【0032】請求項9の発明におけるソフトウェア開発支援方式は、影響範囲の見積とソフトウェア開発の各利用者の担当部分の情報とから、改造の部分の担当者を検索し提示する。即ち、例えばモジュール間のシステム構成上の関係を辿り影響範囲を求め、モジュール毎の担当部門、担当者を上記の統合データベースから検索することで、設計時の変更、問題点の解消のための連絡先、即ち改造が必要な部分の担当者を提示することが可能になる。

【0033】請求項10の発明におけるソフトウェア開発支援方式は、影響範囲の見積と各成果物の工数に基づき、改造の工数の見積を行う。工数の予測を行うことで、改造の規模を把握するための情報が提供でき、改造の方式が複数考えられるような場合に、それらを比較検討するための指標として使用できる。

【0034】請求項11の発明におけるソフトウェア開発支援方式は、利用者が入力する開発対象のシステムの特徴に基づいて、過去の開発事例から開発対象のシステムと関連の深い開発事例を選択し該開発事例の成果物をテンプレートとして使用する。システム、より細かくはモジュールの特徴、即ち機能、性能を検索することで、過去に開発した類似システム、モジュールを提示することが可能になる。予め用意されたテンプレートだけではなく、よく似た事例を検索する機能を用いることにより、新規設計時にも設計の方針を得やすくなる。

【0035】請求項12の発明におけるソフトウェア開発支援方式は、過去の開発事例について非定型資料関連付けステップにおいて管理される開発事例の関連付け情報を参照することにより、開発事例に関連する情報を提示する。開発事例を検索した際には、関連付け情報を辿りその開発事例の設計の根拠を知ることで、新規設計で変更すべき点をより良く把握できる。

【0036】

【実施例】

実施例1. 以下、この発明の一実施例を図について説明する。図1はこの発明の一実施例によるソフトウェア開発支援方式の構成を示すブロック図であり、図において、101はデータベースインタフェイス、108はプロジェクト属性を格納するプロジェクト属性データベース（統合データ、ベース）、109は利用者属性を格納する利用者属性データベース（統合データ、ベース）、110は定型資料を格納する定型資料データベース（統合データ、ベース）、111は事例情報を格納する事例データベース（統合データ、ベース）、112は関連付け情報を格納する関連付けデータベース（統合データ、ベース）、113は非定型資料を格納する非定型資料データベース（統合データ、ベース）である。データベースインタフェイス101は、表示制御部102、属性決定部103、リポジトリ検索部104、事例検索部105、関連決定部106、外部アクセス部107からなる。

【0037】次に動作について説明する。データベースインタフェイス101は、利用者と各データベースとの間のインタフェイスをつかさどる。データベースインタフェイス101に含まれる各サブシステム、表示制御部102、属性決定部103、リポジトリ検索部104、事例検索部105、関連決定部106、外部アクセス部107は、互いに交信しながら各データベースにアクセスし、利用者の要求するデータの呼び出し、更新を行う。

【0038】プロジェクト属性データベース108は、ソフトウェア開発の支援の対象となるソフトウェア開発プロジェクトに関する、開発体制、開発内容等の情報を格納している。具体的には、プロジェクトの名称、プロジェクト責任者、担当者、担当者のチーム分け、チーム毎の担当内容、プロジェクトの開発対象システムの情報が格納される。図2はプロジェクト属性データベース108内に格納されるプロジェクト属性情報の形式を示す図であり、図において、201はプロジェクト番号、202はプロジェクト名称、203は責任者の利用者番号、204は担当者の複数の利用者番号、205は複数のチーム情報、206はソフトウェアシステムのシステム番号である。プロジェクト番号201、利用者番号203、204、システム番号206は、それぞれプロジェクト、利用者、ソフトウェアシステムを識別するためのものであり、この発明によるソフトウェア開発支援方式によるソフトウェアシステムに登録されるプロジェクト、システム、利用者それぞれの通し番号である。尚、プロジェクト属性データベース108は、プロジェクト番号201又はプロジェクト名称202をキーとして検索される。

【0039】図3はプロジェクト属性情報から参照されるチーム情報205の形式を示す図であり、図において、301はチーム番号、302はチーム名称、303はチームの責任者の利用者番号、304は所属する担当者の複数の利用者番号、305～307は担当する複数の製作単位の番号であり、チーム情報205はこれらの情報から成る。チーム番号301はプロジェクト内でのチーム分けを識別するためのものであり、プロジェクト内での通し番号である。製作単位の番号は、プロジェクト内の成果物を識別するためのものであるが、製作単位は適用する開発手法により変化する。例えば、オブジェクト指向分析／設計を適用する場合プログラムは、サブシステム、モジュール、クラスの順に細分化されるので、製作単位の番号の並びは、これらの番号の並びとなる。クラス単位での指定をするためには、図3に示すように、その属するサブシステム番号305、サブシステム内でのモジュール番号306、モジュール内のクラス番号307の組合せで指定する。モジュール単位での指定をする場合には、それが属するサブシステム番号305、サブシステム内でのモジュール番号306とモジュール自身であることを示す識別子で指定する。モジュール自身であることの識別子は、例えばクラスの数番を1から行い0がモジュール自身を差すことにはすれば実現できる。

【0040】利用者属性データベース109は、個々の利用者の担当内容等の情報を格納している。具体的には、利用者の名称、参加するプロジェクト、参加プロジェクト内での所属チーム、担当する製作単位の情報を格納する。図4は利用者属性データベース109における

利用者属性データの形式を示す図であり、図において、401は利用者番号、402は利用者名称、403は複数の参加プロジェクト情報である。また、図5は利用者属性データベース109の利用者属性データから参照される参加プロジェクト情報の並び403の形式を示す図であり、図において、501はプロジェクト番号、502は担当情報の並びである。また、図6は参加プロジェクト情報の並び403から参照される担当情報の並び502の形式を示す図であり、図において、601はチーム番号、602～604は利用者の担当する複数の製作単位の番号、即ちサブシステム番号、モジュール番号、クラス番号である。尚、利用者属性データベース109で使用される利用者番号401、プロジェクト番号501、チーム番号601、製作単位の番号、即ちサブシステム番号602、モジュール番号603、クラス番号604はプロジェクト属性データベース108で使用されるものと共通である。尚、利用者属性データベース109は利用者番号401又は利用者名称402をキーとして検索される。

【0041】定型資料データベース110は、ソフトウェアのシステム開発の各開発フェーズで作成されるシステム構成図、データフローチャート等の定型的な成果物を格納している。具体的には、システム構成、機能設計、詳細設計、ソースコード、オブジェクトコード、実行形式、単体試験仕様、単体試験結果、組合せ試験仕様、組合せ試験結果、総合試験仕様、総合試験結果等テンプレート作成の対象となるシステム開発の各開発フェーズでの成果物全てが定型資料データベース110に格納される。

【0042】例えば、オブジェクト指向分析／設計を行う場合は、サブシステム間の関係を表現するサブシステム交信モデル、オブジェクト間の関係を表現するオブジェクト交信モデル、オブジェクトの状態遷移を表現するオブジェクト状態モデル、オブジェクトの動作とデータのやり取りを表現するアクションデータフローダイアグラム（以下ADFDと略記する）、全体の動作の制御を表現する制御スレッド図等多数の成果物を作成することが定められている。図7はオブジェクト状態モデルの一例を示す説明図、図8はADFDの一例を示す説明図、図9に制御スレッド図の一例を示す説明図である。これらの図は本位田真一／伊藤潔監訳「統・オブジェクト指向システム分析」啓学出版（エス・スレーラ、ジェイ・メラー著、ブレンティスホール）より引用（状態モデルは41頁、制御スレッド図は102頁、ADFDは122頁に記載されている）した。

【0043】定型資料データベース110に格納される定型資料情報にはその改訂履歴を管理するための版情報と、テキスト、ダイアグラム、ブロック図と言った情報の内容に応じて表示方式を決定するための表示制御情報が付加される。図10は定型資料データベース110の

定型資料情報の形式を示す図であり、図において、1001はシステム番号、1002はシステム名称、1003は成果物情報の並びである。図11は図10に示す定型資料情報から参照される成果物情報の並び1003の形式を示す図であり、図において、1101は成果物番号、1102は成果物名称、1103は開発フェーズ番号、1104は製作単位番号、1105は工数、1106は表示識別番号、1107は複数の成果物実体情報、1108は未編集リストである。開発フェーズ番号1103は、開発フェーズ名との対応表において定義される。開発フェーズの分け方は適用するVカーブモデル、スパイラルモデル等の開発手法に応じて定められる。成果物実体情報の並び1107は、成果物の版毎の改訂履歴を示す。工数1105には設計完了前は見積、完了後は実績が入る。未編集リスト1108は当該成果物のテンプレートの既定値が設計値で置き換えていないもののリストである。これは成果物実体情報の形式により異なる。

【0044】例えば、図9に示した制御スレッド図の場合、各プロセスの処理時間が設計対象であるから、未編集リスト1108の初期値はプロセス名が列挙されたものである。設計の進展に伴って、つまり設計値で既定値が置き換えられる毎に未編集リスト1108から当該のものが除かれる。設計が完了した時点では、未編集リスト1108は空になる。尚、定型資料データベース110は、システム番号1001又はシステム名称1002をキーとして検索される。

【0045】図12は図11の成果物実体情報の並び1107を示す図であり、図において、1201は成果物実体番号、1202は版番号、1203は利用者番号、1204はチーム番号、1205は実体呼び出し番号、1206はエントリ情報の並びである。また、図13は図12のエントリ情報の並び1206を示す図であり、図において、1301はエントリ番号、1302は位置情報、1303は関係付け番号である。

【0046】図11に示す表示識別番号1106は、フロック図、アローダイアグラム、テキスト等、用いられる情報の表示形式に応じて番号付けされたものである。また、図13に示す位置情報1302は表示形式により指定の方法が異なる。ブロック図のようにブロック単位で意味を持つものはブロック番号を内容とし、アローダイアグラムのようにノードとアーケークそれぞれが意味を持つものは、ノードとアーケークの種別とその番号を内容とする。図12の実体呼び出し番号1205は資料の実体の呼び出しのための識別子として用いられる。成果物としてのソースコードやオブジェクトコード、実行形式等は通常のファイルシステムのファイルとして格納されてもよいし、特定のCASEツールのリポジトリに加えられてもよい。

【0047】事例データベース111は過去の開発事例

を新たに開発するシステムのひな形として使用するための検索情報を格納している。具体的には、過去に開発されたシステムの名称とそれらの機能、性能の特徴に関する情報を格納する。即ち、システム、モジュール毎にその機能、性能をもとに特徴付けを行い、それらをキーとして過去の開発事例を検索することを目的として、それらの機能、性能が事例データベース111に登録されている。機能、性能は、それらを表現する項目を予め設定しておき、例えば項目毎に数値化して評価することで表現する。図14は事例データベース111における事例データの形式を示しており、図において、1401は事例番号、1402はシステム番号、1403は機能評価値（又は機能特性）の並び、1404は性能評価値（又は性能特性）の並びである。システム番号1402は事例となるシステムを識別するためのものであり、プロジェクト属性データベース108で使用するものと共通である。

【0048】機能評価値（又は機能特性）1403、性能評価値（又は性能特性）1404はプロジェクトの終了毎に各システムを機能、性能で特徴付けるために利用者が評価して定めるものである。機能評価値（又は機能特性）1403はシステムの機能を数値化して表現するものである。例えば、リアルタイム性、汎用度等の項目化を行い各々に対して評価値を与える。また、機能評価値（又は機能特性）1403として適用分野を用いてもよく、アプリケーションの種別による分類を行いその番号を指定することもできる。例えば、データベース検索システムの中でも、検索対象が銀行の口座であるか、資材の在庫であるか、保険の顧客名簿であるか等の分類ができる。あるいは、PATLISのキーワード検索のような方式を取ることも可能である。性能評価値（又は性能特性）1404は機能評価値（又は機能特性）1403と同様に、システムを機能面から特徴付ける項目をあらかじめ設定しておき、それらに評価値を与える。例えば、応答性、モジュールサイズ等の項目を設定した場合はベンチマークによる定量データを使用できる。また、性能評価値又は性能特性1404として、変更の容易性を示す保守性等利用者の主観による評価のほかに、試験結果を不具合の改修の程度をしめすシステムの安定性等も使用できる。尚、事例データベース111は、機能評価値（又は機能特性）1403又は性能評価値（又は性能特性）1404に基づく評価閾数で検索される。

【0049】関連付けデータベース112は、設計内容の裏付けを得たり、過去の開発事例の設計からその試験結果を調べる等、関連情報の検索に使用する関連付けについての情報を格納される。具体的には、定型資料データベース110と、以下で詳細に説明する非定型資料データベース113とに格納される全てのデータの間の関連付けの情報が関連付けデータベース112に格納される。図15は関連付けデータベース112における関連

付け情報の構成を示す図であり、図において、1501は関連付け番号、1502は関連付けの重み付けの値、1503、1507はシステム番号、1504、1508は成果物番号、1505、1509は版番号、1506、1510はエントリ番号である。このように、1つの関連づけ情報には1つの関連づけ番号1501が付与され、1対の関連付けられた成果物のシステム番号、成果物番号、版番号、及びエントリ番号から構成される。

【0050】関連付けの重み付けの値は関連づけられる資料の双方が定型資料であるときにのみ与えられる。関連付け先が非定型資料の場合は、システム番号1503又は1507は非定型資料であることを示すマークとなり、成果物番号1508は非定型資料番号となる。関連付け番号1501は、関連付け情報を識別するための番号であり、全ての関連付けを通して裁番される。システム番号1503、1507、成果物番号1504、1508、版番号1505、1509、エントリ番号1506、1510、非定型資料番号（関連付け先が非定型資料の場合）は、プロジェクト属性データベース108、定型資料データベース110、非定型資料データベース113等のものと共通である。非定型資料であることを示すマークは、例えばシステム番号1503又は1507を1から裁番し、0を非定型資料のマークとして実現できる。尚、関連付けデータベース112は、関連付け番号1501で検索される。

【0051】非定型資料データベース113は定型資料データベース110に蓄えられる成果物の裏付け資料として参照される外部資料を格納している。具体的には、議事録、打合せ資料等、定型資料データベース110に格納される以外の情報が格納される。非定型資料には、表示制御情報が付加される。図16は非定型資料データベース113における非定型資料の形式を示す図であり、図において、1601は非定型資料番号、1602は非定型資料名称、1604は表示識別番号、1605は実体呼び出し番号である。非定型資料番号1601は、定型資料との関連付けられた非定型資料を検索するための識別子であり、それらの非定型資料の通し番号である。実体呼び出し番号1605は、定型資料と同様に実体へのアクセスに使用する。表示識別番号1604も定型資料と同様に、非定型資料の表示形式の識別に用いられる。尚、非定型資料データベース113は非定型資料番号1601又は非定型資料名称1602で検索される。

【0052】次にデータベースインタフェース101の各部における動作について説明する。表示制御部102は、定型資料データベース110及び非定型資料データベース113に格納されている定型資料及び非定型資料の情報の属性に応じて、CRT等の表示装置への適切な表示の制御を行う。表示の形式は表示識別子である表示識別番号1106、1604により決定される。

【0053】属性決定部103は、利用者が使用開始時に入力する利用者名称（図4の利用者名称402に対応）、プロジェクト名称（図2のプロジェクト名称202に対応）、チーム名称（図3のチーム名称302に対応）から、図2に示す形式を有するプロジェクト属性データベース108、及び図4に示す形式を有する利用者属性データベース109を検索して、当該利用者の担当するシステム番号206、製作単位番号等のリストを得る。チーム名称を入力するのは、複数のチームに重複して所属することに対応するためである。成果物の版は最新を仮定するが、特に指定されたときは、指定された版の値を使用する。

【0054】リポジトリ検索部104は、属性決定部103が得たシステム番号206、モジュール番号306、版番号1202から、定型資料データベース110を検索し、利用者が行う作業の対象となる成果物実体番号1201を得る。

【0055】事例検索部105は、利用者が入力するシステムの機能、性能の情報から、事例情報データベース111を検索し、それらのパラメータがもっとも近いシステム番号1402を得る。パラメータのマッチングには任意の方法を用いてよい。例えば、あらかじめ入力されたパラメータと事例情報データベース111内のパラメータを用いた評価式をいくつか用意しておき、検索時にそれらからの選択を許したり、検索時に評価式を自由に作成させる等の方法がある。いずれの場合も、評価式の最大化又は最小化を行い、上位のいくつかを結果として出力する。

【0056】関連決定部106は、利用者が定型又は非定型資料の表示上で選択する関連呼び出し情報から、関連付けデータベース112を検索し、関連付けの他方の成果物番号1504又は1508、即ち定型又は非定型資料番号を得る。外部アクセス部107は、定型又は非定型資料のうち、実体が外部のデータベースに蓄えられたものについて、それが蓄えられたデータベースにアクセスし、当該データを呼び出す。

【0057】次にこの実施例のソフトウェア開発支援方式によるソフトウェア開発の動作について説明する。ソフトウェアシステムの開発に先立ち、各開発フェーズにおいて生成すべき成果物のガイドラインをテンプレートにより定めておく。テンプレートは、過去に開発した類似システムの平均的なシステム構成と、それに含まれる個々のモジュールの機能、性能とを含む。クラス、モジュール間でのデータのやり取り、関数、ロジックの呼び出し等の論理的な関係はシステム構成に含まれる。各開発フェーズ内では、システムのモジュール構成等の情報に基づき、成果物は相互に関連づけられ、その関連付け情報が上記した関連付けデータベース112に格納されている。これ以外に、設計者が着目する成果物の間の関連づけを自由に行えるように、全ての開発フェーズをま

たがり成果物間の関連付け情報が関連付けデータベース112に格納され得る。関連づけはシステム開発中に生成される全ての書類を対象とする。仕様を決定するための会議の打合議事録、開発チームの間での通信記録等も関連付けデータベース112に取り込み、関連づけの対象とする。モジュール間の関連づけには相互の関連の強さによる重み付けを行う。関連付け情報は、モジュール間の関数の呼び出し等論理的な構造により決定されるが、重み付けを得るためのそれらの客観的な指標が得られないときには、利用者により決定される。重み付けは成果物の間に限られる。外部資料との関連付けには重み付けは行わない。

【0058】図17、図18、及び図19はそれぞれ関連付けデータベース112の関連付け情報を得る処理の流れを示すフローチャートである。成果物間の関連付けにはモジュール間の関数等の呼び出し関係等論理的に定まるものと、利用者が自由に設定するものとの2通りがある。前者は、市販のCASEツールに含まれるモジュール構成エディタ等でシステム設計を行う過程でその関連付け情報は自動的にCASEツールのデータベースに登録されるので、その関連付け情報から自動的に生成して関連付けデータベース112に格納することができる。重み付けについては関数間呼び出し等に基づく依存関係から自動的に定めることができるが、利用者による上書きを行うことも可能である。

【0059】他方、後者の場合は、図17に示すように、ステップST1において、まず利用者は関連付けを行いたい2つの資料を選び、各々の任意の位置を指定することによりエントリを設定し両者を関連付ける。ステップST2において、両方（又は両端）のエントリを設定したか否かを判定し、まだ両方のエントリを設定していないならばステップST1に戻る。次に、ステップST3において、関連付けの両方が定型資料か否か調べ、関連付けの両方が定型資料である場合は、ステップST4において、関連付けの重み付けの値が利用者により設定される。そして、ステップST5において、作成した関連付け情報を登録して終了する。

【0060】図17におけるステップST1のエントリ設定は図18のフローチャートに従って実行される。まず、ステップST11において、利用者は関連付けを行いたい2つの資料における各々の任意のエントリ位置を設定する。自動生成による関連付けと利用者による関連付けのいずれの場合も、ステップST12において、指定された位置にエントリがあるか否かを判定し、エントリがないときはステップST13及びST14において新たにエントリを作成し登録する。また、図18のステップST11におけるエントリ位置設定は図19に示すフローチャートに従って実行される。エントリの位置指定は、まず、ステップST21において、関連づけられる資料の任意の部分がブロック図のような構造データか

否かを判定し、ブロック図のような構造データの場合は、ステップST22においてブロック等の要素を指定し、テキストデータのような非構造データの場合は、ステップST23において文頭からの位置等座標で指定する。

【0061】関連付けデータベース112は、打合せ議事録、打合せ資料、参考文献等ソフトウェア開発支援方式が直接関与しない非定型資料と、ソフトウェア開発支援方式を経由して生成される定型資料である成果物との間の関連付け情報を管理できるようにしてよい。この場合、図19に示すエントリの位置指定をグラフィックイメージ等に対応するように修正した、図20に示すエントリ位置設定のフローチャートで実現される。打合せ議事録、打合せ資料、参考文献等ソフトウェア開発支援方式が直接関与しない非定型資料は、写真やFAXの出力等のイメージを含むために、2次元での位置指定が必要になる。まず、ステップST31において、関連づけられる資料の一方が定型資料であるか否かを判定し、定型資料でないならば、ステップST32において、関連づける情報の位置を2次元で指定する。定型資料である場合の処理（ステップST33～ST35）は、図19に示すとおりであるので重複説明は省略する。

【0062】以上述べたように、全ての成果物を統合データベースに格納する際に、他の製作単位や他の開発フェーズとの関連付け情報が関連付けデータベース112に記録される。関連付け情報にはシステムの構成により定まるものと、利用者が自由に設定するものとの2種類あり、全ての成果物は前者の第1のタイプにより相互に関連付けられて保存されている。成果物のいずれかを修正する場合は、その修正内容に応じて関連する成果物を修正し全体の整合を取る必要があるが、関連付けデータベース112に格納された関連付け情報がその助けになる。即ち、成果物の間での関連付けを行うことで、あるモジュールを変更する際に関連するモジュールを検出し提示することが可能になる。この際、後者の第2のタイプの利用者が設定した関連付け情報を辿り背景となる設計の根拠を知ることで、より正確な判断が下せる。また、関連付け情報は、利用者が関連付けられたモジュールを確認するために呼び出すためにも使用できる。さらに、関連付けが議事録等の非定型資料にも及ぶことで、設計内容の背景の確認が容易になり、設計方針の誤った解釈によるミスを減少させることができる。

【0063】次に各開発フェーズにおいて生成すべき成果物のガイドラインを定めるテンプレートについて説明する。機能設計開発フェーズのテンプレートは各モジュールの機能の記述を含み、詳細設計開発フェーズのテンプレートは各モジュールの性能の記述を含む。個々のモジュールの機能、性能は、従来のシステムの平均的な内容を既定値として設定する。ソフトウェアシステムの開発は、テンプレートに埋め込まれた既定値を、新たに設

計した内容で置き換える方式で行う。テンプレートの例としては、図7や図8や図9の様な定型資料についての典型的な構成と内容を持つものである。これらに含まれる内容は規定値として用いられる。新たな設計内容で規定値を置き換えることで、ソフトウェアシステムの設計を行うことになる。

【0064】ソフトウェア開発の各開発フェーズ毎にソフトウェアの機能、性能の検証を行うにあたり、上記したテンプレートの既定値を用いて模擬動作を実行し、部分的な設計内容がシステム全体に要求される機能、性能に合致するかどうかを検証する。各開発フェーズでは、当該開発モジュール毎に記述された機能、性能の内容に従って動作の模擬を行う。機能設計開発フェーズでは、システムの動作内容の模擬を行い、詳細設計レベルでは、応答性等性能に関する模擬を行う。かかる機能、性能の検証は、図7、図8、図9に示した様な定型資料のテンプレートを使用して行う。部分的に設計の行われた成果物は一部の設計内容で既定値が置き換えられている状態である。

【0065】図8に示したようなADFには、データのやり取りにどのような変換が行われるかが記述される。ADFはシステム外部との入出力も記述されるので、ADF上でのシステム外部からの入力からシステム外部への出力までの全てのバスを検証することで、システムが正しくデータを加工しているかどうかを判定できる。これによりシステムの機能の検証が行える。この機能の検証はデータの流れをトレースすることで行うため、検証の対象となるモジュールについて全てのデータの授受が記述されている必要がある。

【0066】性能の検証は、図9に示した制御スレッド図のように、プロセス毎の動作制御を記述した図を用いて行われる。ここで言うプロセスとはデータの書き換えや式の計算等、意味のある処理の単位である。制御スレッド図の各プロセスにその処理に必要な時間が添えられている。各プロセスはそれに先行する全てのプロセスが完了するまで処理を開始できない。制御スレッド図を上から下にたどりながら、所要時間を集計することで対象モジュールの応答性能が評価できる。制御スレッド図もADFと同様、すべてのプロセスについての処理時間、前後関係が記述されている必要がある。

【0067】前述のように設計途上のシステムについては、テンプレートの未設計の部分については既定値が埋め込まれている。設計が完了していない部分については既定値を用いてトレースすることで、設計が途中のモジュールについても機能、性能の検証が行える。テンプレートの規定値を用いてソフトウェア開発の各開発フェーズ毎に、システムを構成する要件が全てそろっているかどうかを判定することができる。設計作業はテンプレート上の既定値を新たな設計値で置き換えるか、新たな要素を追加することで行われる。テンプレートの呼び出し

時に各定型資料の要素すべてを、図11に示す成果物情報1003に含まれる未編集リスト1108に登録する。既定値の置き換え毎に当該要素を未編集リスト1108から取り除く。未編集リスト1108に要素が残っているかどうかが、システム設計完了の判定になる。より明確に言うと、ソフトウェア開発の進捗を既定値が設計値ないし設計内容で置き換えた比率で把握する。既定値を設計値で置き換える動作を行ったときに、そのモジュールの設計が済であることのマーク付けを行う。モジュールによっては既定値をそのまま使用する場合もあるが、この場合も設計が済であることを示すマーク付けを行う。各モジュールに付けられた設計が済であるか否かのマークを集計することで、システム開発の進捗を把握するための指標を提示することが可能になる。

【0068】以上述べたように、既定値を持つテンプレートを使用することで、システム設計の際の指針を利用者に与えることができ、設計の効率を向上させることができる。また、テンプレートには設計時の漏れを防ぐ効果もある。さらに、既定値の埋め込まれたテンプレートを使用することで、個別モジュールの内容を検討する際にそのモジュールについては新たに設計した内容を用いて、残りのモジュールについては既定値を用いて動作の模擬を行うことにより、設計途上のシステムのためのテンプレート上での機能、性能の検証を行うことができ、部分的な設計がシステム全体の要求に合致しているかを確認できる。従って、全体の設計が終わってから機能性能を見積るのに比べて早い時期での修正が行うことができ、設計上の効率を増すことができる。

【0069】次に版管理について説明する。ソフトウェアをサブシステム、モジュール等に細分化するとき、その各々についての版とそれらを組み合わせた全体の版管理を行うとともに、関連付けの改訂履歴を含めた版管理を行う。図12の定型資料データベース110の成果物実体情報1107の構造に示したように成果物実体と版番号1202とを組にして管理することで行う。成果物実体が修正されて新たな版として登録されたときは、当該成果物実体を指定された版番号1202と組合せて新たな定型資料として登録する。関連付けが変更されたときも新たな版として登録する。関連付けの変更による版更新は、図17に示した関連付け登録のフローチャートに図21に示したように版更新の処理を追加することで実現される。図21に示すように、ステップST41において、まず利用者は関連付けを行いたい2つの資料を選び、各々の任意の位置を指定することによりエントリを設定し両者を関連付ける。ステップST42において、両方（又は両端）のエントリを設定したか否かを判定し、まだ両方のエントリを設定していないならばステップST41に戻る。次に、ステップST43において、作成した関連付け情報を登録し、ステップST44において、その関連付け情報に関連する成果物等の版を

更新して終了する。版の呼び出しは、プロジェクト属性データベース108、利用者属性データベース109及び定型資料データベース110を検索して成果物一覧を引き出し、各成果物の最新の版を呼び出すことで行う。特定の版の呼び出しの場合は、指定された版より少ない値で最大の版番号を持つ版を呼び出す。

【0070】このように、最小単位毎に版管理を行うことで、関連付けの単位と版管理の単位とが一致することになり、関連付けを版毎に管理することが可能になる。即ち、版情報を実体と分離して管理することで、最小の製作単位毎の柔軟な版管理が実現でき、外部から取り込まれた非定型資料等のデータに対しても関連付けが行えるようになる。

【0071】次に、図22に示す利用者の関連情報の呼び出し処理のフローチャートを参照しながら、利用者が利用を開始するときに入力する情報に基づいて利用者属性データベース109を含む統合データベースから必要な情報を呼び出す動作について説明する。既に述べたように、利用者属性データベース109は利用者の参加するプロジェクト、プロジェクト内のチーム、担当する部分等に関する情報を管理している。図22に示すように、ステップST51において、利用者名が入力されると、ステップST52において、属性決定部103が、図4に示す利用者属性データベース109の利用者情報から当該利用者の参加するプロジェクトの情報を得てプロジェクトリストを作成し、ステップST53において、そのプロジェクトリストをCRTディスプレイ等の表示装置に表示する。次に、ステップST54において、利用者がそのプロジェクトリストの中から特定のプロジェクトを選択すると、ステップST55において、当該担当者の担当する担当内容を決定し制作単位の成果物実体を呼び出す。

【0072】従って、利用者が当該ソフトウェア開発支援方式の使用開始時に入力する情報をプロジェクト、利用者の属性と照合し、当該作業に必要な情報を検索、収集して利用者に提示することが可能になる。次に、開発中または開発済みのシステムの一部又は全部の機能を変更するにあたって、システムの成果物の間のシステム構成上の関係に基づいて、変更内容に直接関与する部分の改造が影響を及ぼす範囲の見積処理について説明する。図23は影響範囲の見積処理の流れを示すフローチャートであり、以下この図を参照しながら説明する。まず、ステップST61において利用者が変更対象を選択し改造対象が決定すると、ステップST62において、システム構成上の成果物間の関連付け情報を関連付けデータベース112から検索し改造による影響を受ける可能性のある制作単位を求める。そして、ステップST63において、製作単位毎に改造の及ぼす影響度を計算して、ステップST64において、算出した影響度から影響範囲を決定する。

【0073】システム構成上の関連付け情報の表現方式はシステム設計の方式、実装言語によって違うが、ここではオブジェクト指向分析／設計に基づき、C++等のオブジェクト指向言語による実装を行う場合について説明する。C++の場合、実装の最小単位はクラスである。クラスの間の関連付け情報は、関数呼び出し、データメンバのアクセス、継承関係である。これらの関連付け情報は、クラスをノードとし、関数呼び出し、データメンバのアクセス、継承関係をアーケードとする有向グラフで表現できる。前述のように、関連付け情報には、論理構造から定まるか又は利用者により決定される依存関係の強さの重み付けの値が付加される。アーケードにはこの重み付けの値がラベリングされる。以下この有向グラフを用いてオブジェクト指向言語における影響度の計算について説明する。

【0074】改造にあたり直接修正の対象となるクラス群に対応するノードの集合を原始集合とする。アーケードの向きを考慮して有向グラフを探索し、原始集合のノードから辿れるノードが影響の及ぶ範囲である。ただし、単純にシステム構成を現す有向グラフ上でこのような探索を行うと、システム内の要素は互いになんらかの関連を持つので、全てのノードが影響範囲に含まれる可能性が高くなる。そこで、各ノードが影響範囲内にあるかどうかを、原始集合からの経路の長さと、原始集合からの経路の数を元にして判定する。判定には次の式で表現される評価関数を用いる。

【0075】

【数1】

$$U(v) = \sum_{u \in N} f(t) P((u, v)) U(u) \dots (1)$$

$$u, v \in V, (u, v) \in E$$

【0076】ここで、Vはノードの集合、Eはアーケードの集合である。tは継承であるか、参照であるかといった関連付けタイプ、fはタイプによる重み付けの関数である。Pはvとuの関係に基づいて決まる影響の波及度である。この値は、先に述べた利用者により設定される依存関係の強さである。この評価関数は原始集合からアーケードをたどりながら再帰的に計算される。各ノードの評価関数が大きい値より小さくなったりときは、そのノードの評価関数の値を0とする。これにより、0以外の評価関数の値を持つノードが、影響を受けると予測されるノード群になる。

【0077】以上のように方法で決定されたノードの集合に対応する成果物単位の集合が、改造の結果影響を受ける可能性が高いものであると決定される。影響を受けると予測されるクラス、モジュールが決定されると、影響を受けるモジュールの担当者、担当チームが検索でき、改造の内容を連絡すべき相手先として利用者に提示できる。図24はかかる担当者の検索処理の流れを示すフローチャートである。このフローチャートは図23の改造の影

響範囲の見積処理の流れを示すフローチャートに担当者検索を追加したものである。この図24に示すように、上記した図23のステップST61～ST64と同一なステップST71～ST74で改造の影響範囲が決定されると、ステップST75において、求めた改造の影響範囲に含まれる成果物単位毎に、図12に示す定型資料データベース110の成果物実体情報から担当の利用者の情報を検索して獲得する。

【0078】以上述べたように、システム構成による関連情報をたどるだけでは、全ての成果物が修正の対象になる可能生が高いが、関連付けの強さを反映する評価関数に基づき影響の度合いを評価することで、特定モジュールの変更のシステム全体への影響の度合いを見積るために評価基準を提示することができる。これにより、個々の成果物についての修正の必要の有無の指針が得られ、個別に判定するよりも効率良くまた漏れなく修正対象を決定できる。また、モジュール間の関連付けを辿り、モジュール毎の担当部門、担当者を定型資料データベース110から検索することで、設計時の変更、問題点の解消のための連絡先又は交渉先を提示することが可能になる。

【0079】次に、改造の影響範囲と各成果物の工数に基づく改造の工数の見積処理について説明する。改造による工数の見積は、図23に示した影響範囲の見積処理と、各成果物毎の工数の見積又は実績とを用いて行われる。図25はかかる工数計算処理の流れを示すフローチャートである。図25に示すように、上記した図23のステップST61～ST64と同一なステップST81～ST84で改造の影響範囲が決定されると、ステップST85において、求めた改造の影響範囲に含まれる成果物単位毎に改造の工数を求める。ある変更に伴う改造の工数は各クラスの工数c(v)を用いて

【数2】

$$C(R) = \sum_{v \in N} U(v) c(v) \dots (2)$$

と表現できる。U(v)は式(1)で求めたクラス毎の評価関数、即ち影響波及度である。改造の工数の見積は複数の改造案に対する評価の指標として利用者に提示される。

【0080】上記の改造の影響範囲の決定処理等においては、クラスを対象として記述したが、モジュールが対象であってもよく、モジュールを中心と考えるときは、モジュールとクラスとの包含関係をもとにして、クラス間の関連に基づく有向グラフを縮約して得られる有向グラフを用いて同様のことが行える。

【0081】以上述べたように、改造の工数の予測を行うことで、仕様変更に対して適切な判断を行い改造の規模を把握するための情報を提供でき、改造の方式が複数考えられるような場合に、それらを比較検討するための指標として使用できる。

【0082】次に、ユーザが入力するシステムの特徴に基づいて、統合データベースに蓄えられた過去の開発事例から、関連の深い事例を選択し提示して、新規開発のためのテンプレートとして使用することを可能にする機能について説明する。図26はかかる過去の開発事例から類似システムを得る処理の流れを示すフローチャートである。図26に示すように、まずステップST91において、利用者がシステムの特徴パラメータを入力すると、ステップST92において、事例検索部105が、図14の事例データベース111に格納された過去の事例にはそのシステムの機能、性能の評価のパラメータとして付加された機能評価値1403及び性能評価値1404を用い、ユーザの入力した特徴パラメータとこれらのパラメータとの相関を評価関数により計算する。評価関数は用途に応じて任意のものを用いてよく、例えば項目毎の差の二乗の総和で評価する最小2乗誤差等を用いれば平均的な類似度を評価できる。特定の性質を重視したいときは、パラメータ毎に荷重係数をかける等して調整できる。そして、ステップST93において、評価関数がある閾値より大きいか否かをもって類似システムを決定する。

【0083】以上述べたように、テンプレートだけではなく、システム、モジュールの機能、性能の評価のパラメータを用いて良く似た過去の開発事例を検索して、過去に開発した類似システム、モジュールを提示でき、利用者は新規設計時にも設計の方針を得やすくなる。過去の開発事例に対して設計根拠を提示する機能を有する。過去の開発事例の根拠の提示は、検索した過去の事例からそれに関連づけられている非定型資料を呼び出すことで実現できる。これは、関連決定部106が関連付けデータベース112を検索し関連づけられた過去の事例の非定型資料番号を得、この非定型資料を加工することなく、例えば外部アクセス部107が現在編集途上の成果物と同じ形式で呼び出すことにより可能になる。したがって、過去の事例に関連づけられている非定型資料の呼び出しは、過去の事例の情報をすべてコピーし、図11に示した成果物情報に含まれる編集の進展を示す未編集リスト1108にすべての編集対象をリストアップすることにより実行される。

【0084】以上述べたように、開発事例を検索したときに、利用者が自由に設定した成果物間の関連付けをたどりその過去の開発事例の設計の根拠を知ることで、新規設計で変更すべき点をより良く把握できる。

【0085】

【発明の効果】以上のように、請求項1の発明によれば、各開発フェーズで得られる全ての成果物を相互に関連付ける関連付け情報を格納する統合データベースを設けたものであるように構成したので、関連づけは利用者が関連づけられたモジュールを確認するために呼び出すためにも使用できる効果がある。

【0086】請求項2の発明によれば、少なくとも機能設計及び詳細設計の開発フェーズにおいて、既定値を含むテンプレートを用いて開発を行うものであるように構成したので、これにより、システムの設計の際の指針を利用者に与えることができ、設計の効率を向上させ得る。また、テンプレートの利用は設計時の漏れも防ぐことができる効果がある。

【0087】請求項3の発明によれば、各開発フェーズの直接の成果物以外の資料と、各開発フェーズにおいて生成される成果物とを相互に関連付ける関連付け情報を統合データベースにおいて管理するものであるように構成したので、このように、関連づけが打合せ議事録等にも及ぶことで、設計内容の背景の確認が容易になり、設計方針の誤った解釈によるミスを減少させることができるのである。

【0088】請求項4の発明によれば、サブシステム、モジュールの各々についての版とそれらを組み合わせた全体の版との管理を行い、且つ、関連付け情報の改訂履歴を含めた版管理を行うものであるように構成したので、最小単位毎に版管理を行うことで、関連付けの単位と版管理の単位が一致することになり、関連付けを版毎に管理することが可能になる。関連付け情報を実体と分離して管理することで、柔軟な関連付けの管理が行え、外部から取り込まれたデータに対しても関連付けが行える効果がある。

【0089】請求項5の発明によれば、既定値を含むテンプレートを用いて、ソフトウェア開発の各開発フェーズ毎にソフトウェアの機能、性能の検証を行うものであるように構成したので、既定値の埋め込まれたテンプレートを使用することで、個別のモジュールの内容を検討するにあたって、そのモジュールについては新たに設計した内容を用いて、残りのモジュールについては既定値を用いて動作の模擬を行う可能になる。テンプレート上で機能、設計の見積を行うことにより、部分的な設計がシステム全体の要求に合致しているかを確認できる効果がある。

【0090】請求項6の発明によれば、ソフトウェア開発の各開発フェーズ毎に、テンプレートの既定値が設計値で置き換わった割合によりシステムを構成する要件が全てそろっているか否かを判定するものであるように構成したので、より具体的には、各モジュールに付けられた設計が済であるか否かのマークを集計することで、システム開発の進捗を把握するための指標を提示することができる効果がある。

【0091】請求項7の発明によれば、利用者が利用を開始するときに入力する情報に基づいて、統合データベースから利用者の担当する成果物を呼び出すものであるように構成したので、従って、利用者がシステムの使用開始時に入力する情報をプロジェクト、利用者の属性と照合し、当該作業に必要な情報を検索、収集して提示で

きる効果がある。

【0092】請求項8の発明によれば、システムの成果物の間のシステム構成上の関係に基づいて、開発中または開発済みのシステムの一部又は全部の機能変更に直接関与する部分の改造が影響する範囲の見積るものであるように構成したので、従って、特定モジュールの変更のシステム全体への影響の度合いを見積り、仕様変更に対する評価基準を提示することができる効果がある。

【0093】請求項9の発明によれば、影響範囲の見積とソフトウェア開発の各利用者の担当部分の情報から、改造の部分の担当者を検索し提示するものであるように構成したので、即ち、例えばモジュール間のシステム構成上の関係を辿り影響範囲を求め、モジュール毎の担当部門、担当者を上記の統合データベースから検索することで、設計時の変更、問題点の解消のための連絡先、即ち改造が必要な部分の担当者を提示することができる効果がある。

【0094】請求項10の発明によれば、影響範囲の見積と各成果物の工数とに基づき、改造の工数の見積を行うものであるように構成したので、工数の予測を行うことで、改造の規模を把握するための情報が提供でき、改造の方式が複数考えられるような場合に、それらを比較検討するための指標として使用できる効果がある。

【0095】請求項11の発明によれば、利用者が入力する開発対象のシステムの特徴に基づいて、過去の開発事例から開発対象のシステムと関連の深い開発事例を選択し該開発事例の成果物をテンプレートとして使用するものであるように構成したので、システム、より細かくはモジュールの特徴、即ち機能、性能を検索することで、過去に開発した類似システム、モジュールを提示することが可能になる。予め用意されたテンプレートだけではなく、良く似た事例を検索する機能を用いることにより、新規設計時にも設計の方針を得やすくなる効果がある。

【0096】請求項12の発明によれば、過去の開発事例について非定型資料関連付けステップにおいて管理される開発事例の関連付け情報を参照することにより、開発事例に関連する情報を提示するものであるように構成したので、開発事例を検索した際には、関連付け情報を辿りその開発事例の設計の根拠を知ることで、新規設計で変更すべき点をより良く把握できる効果がある。

【図面の簡単な説明】

【図1】 この発明の一実施例によるソフトウェア開発支援方式の構成を示すブロック図である。

【図2】 上記実施例におけるプロジェクト属性データベース108に格納されるプロジェクト属性情報のデータ構造を示す図である。

【図3】 図2に示すプロジェクト属性情報のうちのチーム情報部分のデータ構造を示す図である。

【図4】 上記実施例における利用者属性データベース

109に格納される利用者情報のデータ構造を示す図である。

【図5】 図4の利用者属性情報の参加プロジェクト情報部分のデータ構造を示す図である。

【図6】 図5の参加プロジェクト情報の担当情報部分のデータ構造を示す図である。

【図7】 オブジェクト指向設計によるソフトウェア開発を行う場合の成果物の一例である、オブジェクトの状態変化を記述したオブジェクト状態モデルの一例を示す図である。

【図8】 オブジェクト指向設計によるソフトウェア開発を行う場合の成果物の一例である、プロセス毎の処理内容とデータの受渡しを記述したアクションデータフローダイアグラムの一例を示す図である。

【図9】 オブジェクト指向設計によるソフトウェア開発を行う場合の成果物の一例である、プロセス毎の処理の制御を記述した制御スレッド図である。

【図10】 上記実施例における定型資料データベース110に格納される定型資料情報のデータ構造を示す図である。

【図11】 図10の定型情報データベース110の成果物情報1003のデータ構造を示す図である。

【図12】 図11の成果物の実体情報1107のデータ構造を示す図である。

【図13】 図12の成果物の実体情報のうちのエンティ情報1206のデータ構造を示す図である。

【図14】 事例データベース111に格納される事例データのデータ構造を示す図である。

【図15】 関連付けデータベース112に格納される関連付けデータのデータ構造を示す図である。

【図16】 非定型資料データベース113に格納される非定型資料データのデータ構造を示す図である。

【図17】 関連付けデータベース112に格納する成果物間の関連付け情報の作成処理の流れを示すフローチャートである。

【図18】 図17に示すフローチャートにおけるエンティ設定処理の流れを示すフローチャートである。

【図19】 図18に示すフローチャートにおけるエンティ位置設定処理の流れを示すフローチャートである。

【図20】 ソフトウェア開発支援方式が直接関与しない資料への関連付け処理におけるエンティ位置設定処理の流れを示すフローチャートである。

【図21】 関連付けデータベース112に格納する成果物間の関連付け情報の作成処理に伴う版管理の流れを示すフローチャートである。

【図22】 利用者の必要とする情報を呼び出す利用者関連情報呼び出し処理の流れを示すフローチャートである。

【図23】 ソフトウェア変更時の改造の影響範囲を見積る影響範囲見積処理の流れを示すフローチャートである。

る。

【図24】 改造の際に改造部分の担当者を検索する担当者連絡処理の流れを示すフローチャートである。

【図25】 改造のための工数の見積を行なう工数見積処理の流れを示すフローチャートである。

【図26】 ソフトウェア開発の際に関連する過去の開発事例を呼び出す事例利用開発処理の流れを示すフローチャートである。

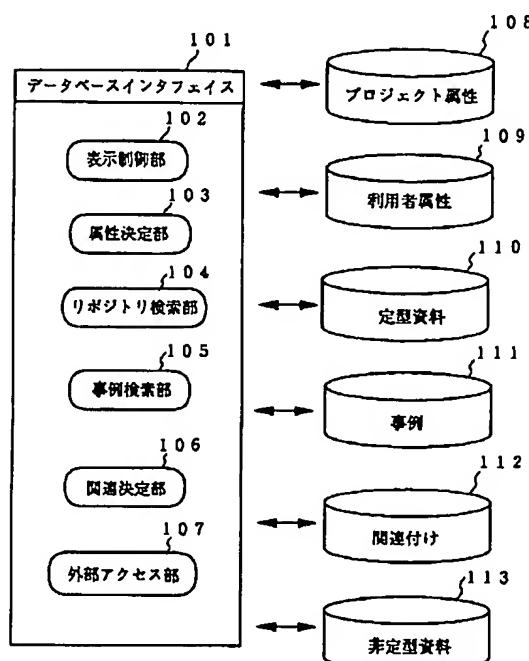
【図27】 従来のソフトウェア開発支援方式におけるソフトウェアプロセスモデルの一例であるVカーブモデル

ルの開発の流れを示す図である。

【符号の説明】

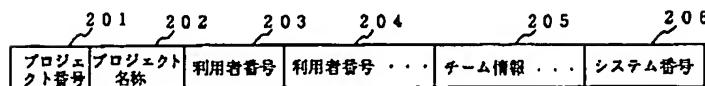
108 プロジェクト属性データベース（統合データベース）、109 利用者属性データベース（統合データベース）、110 定型資料データベース（統合データベース）、111 事例データベース（統合データベース）、112 関連付けデータベース（統合データベース）、113 非定型資料データベース（統合データベース）。

【図1】

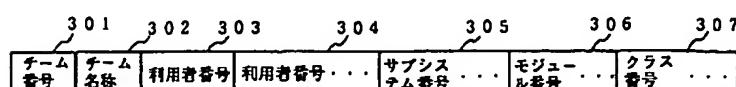


108 : プロジェクト属性データベース（統合データベース）
 109 : 利用者属性データベース（統合データベース）
 110 : 定型資料データベース（統合データベース）
 111 : 事例データベース（統合データベース）
 112 : 関連付けデータベース（統合データベース）
 113 : 非定型資料データベース（統合データベース）

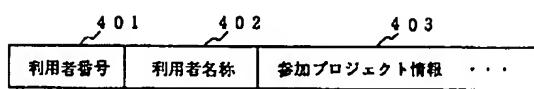
【図2】



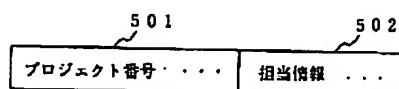
【図3】



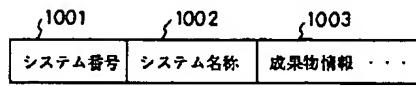
【図4】



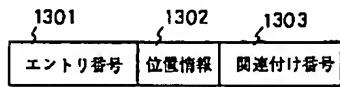
【図5】



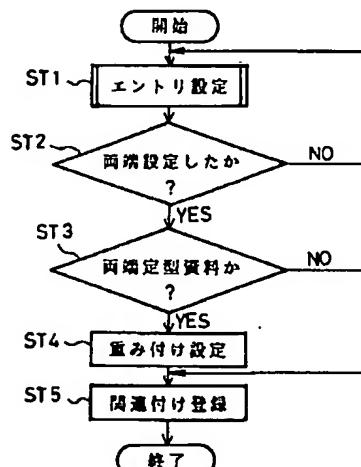
【図10】



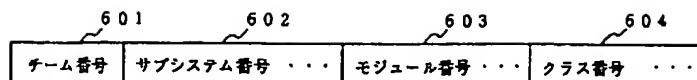
【図13】



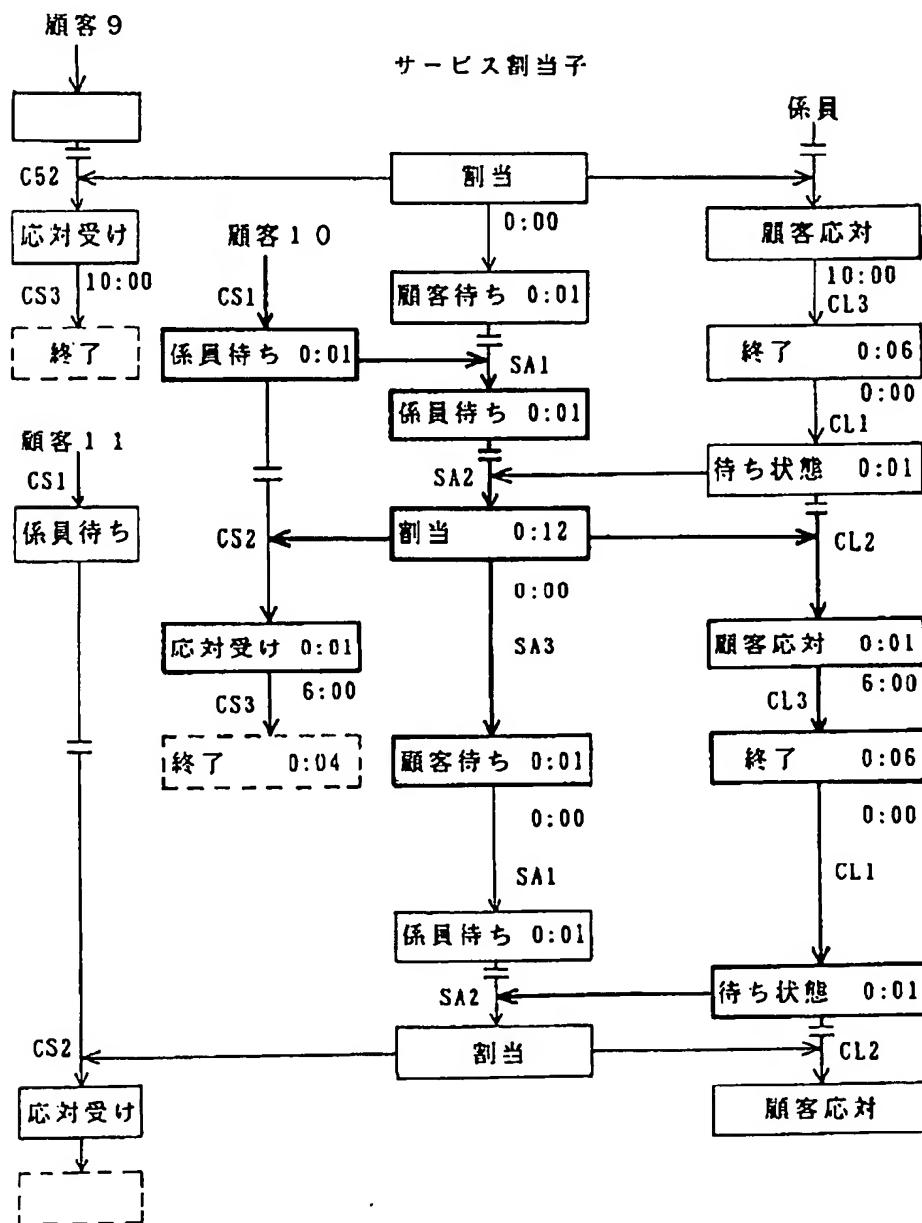
【図17】



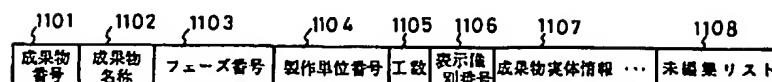
【図6】



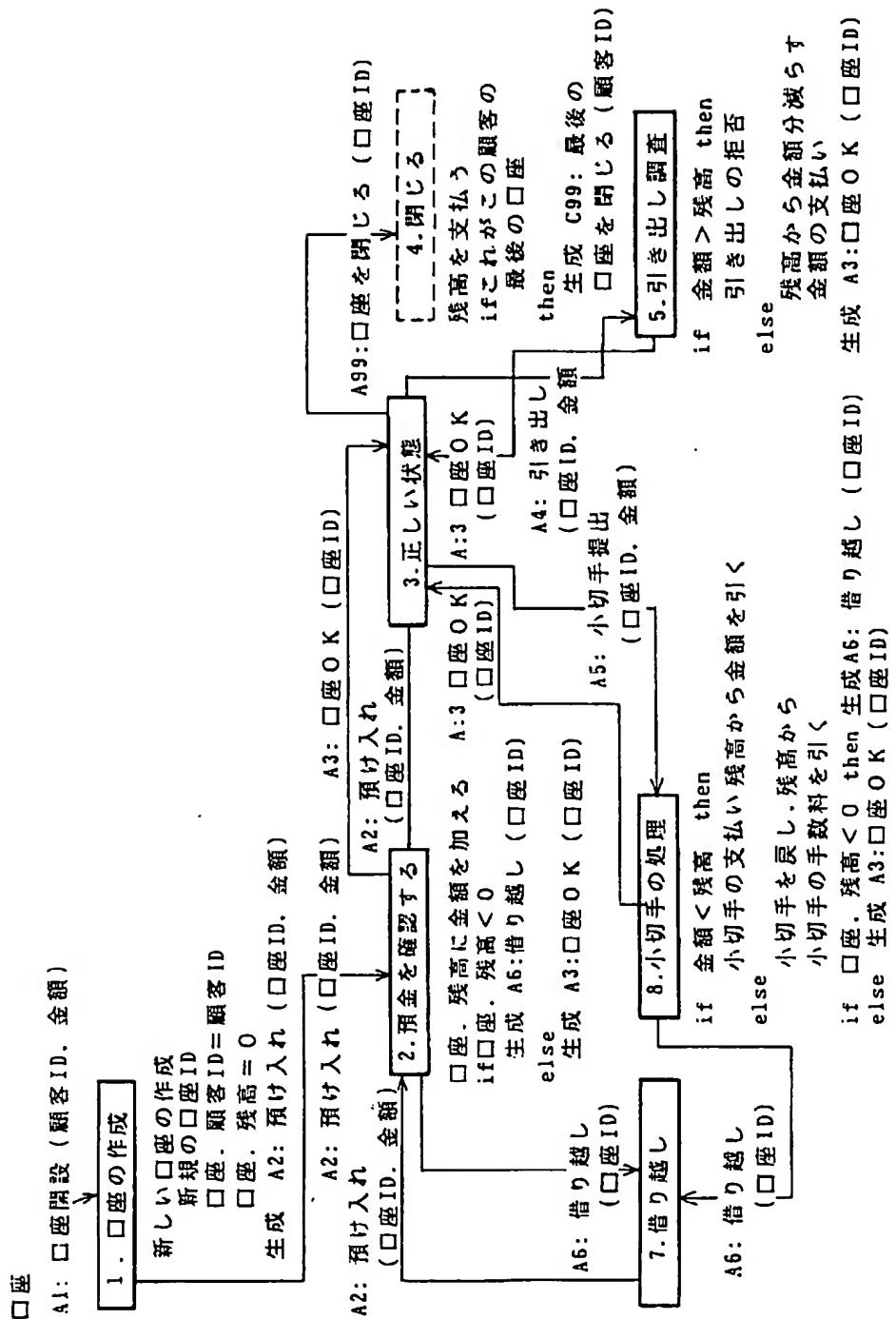
【図9】



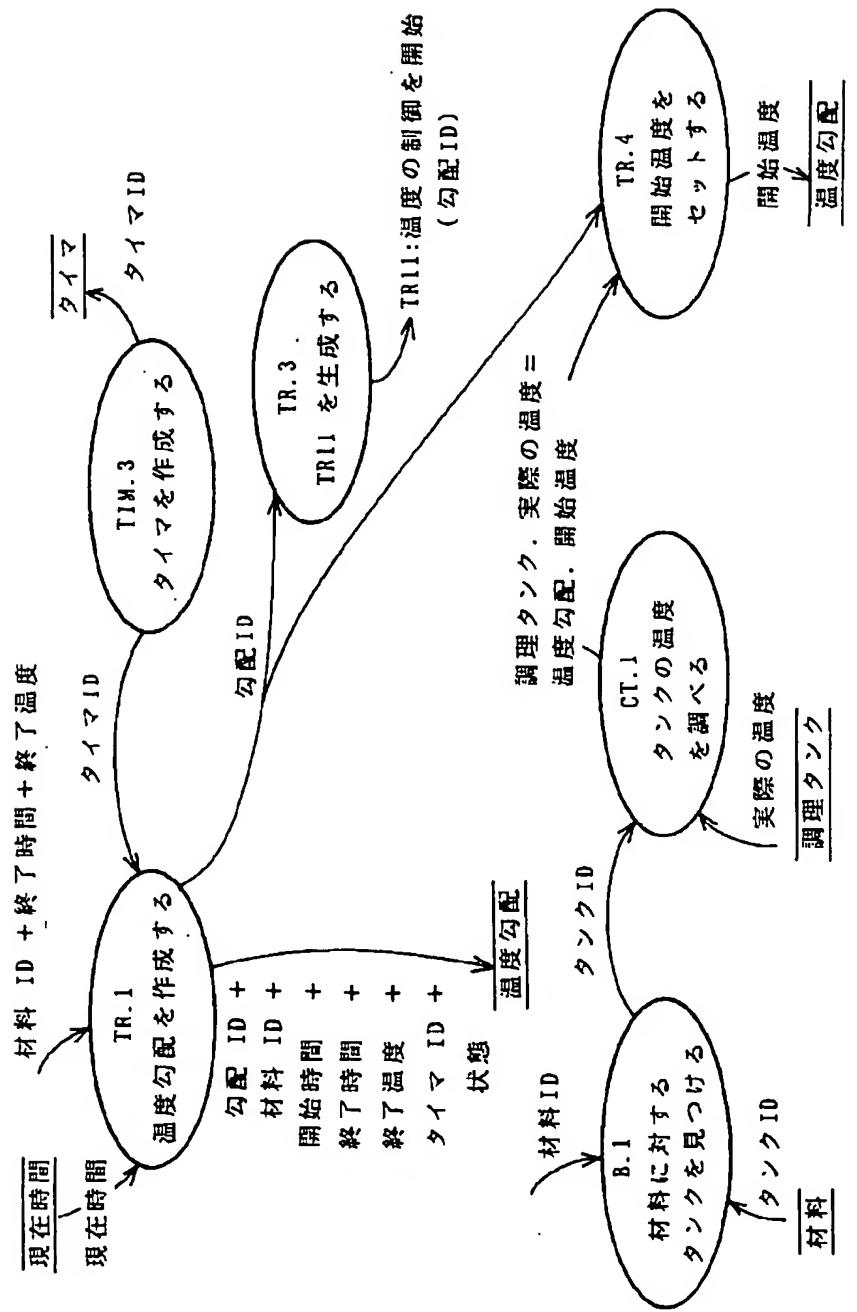
【図11】



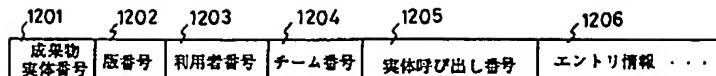
(図7)



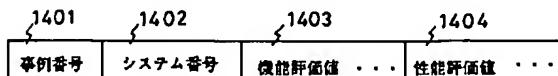
〔図8〕



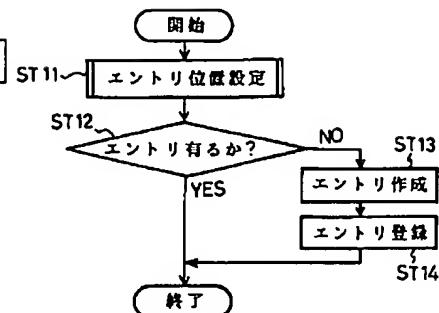
【図12】



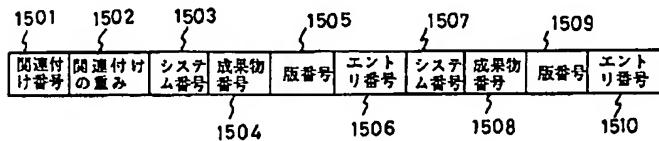
【図14】



【図18】



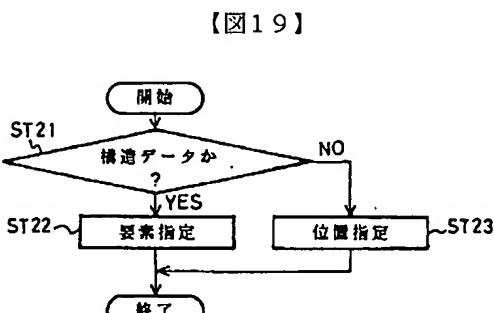
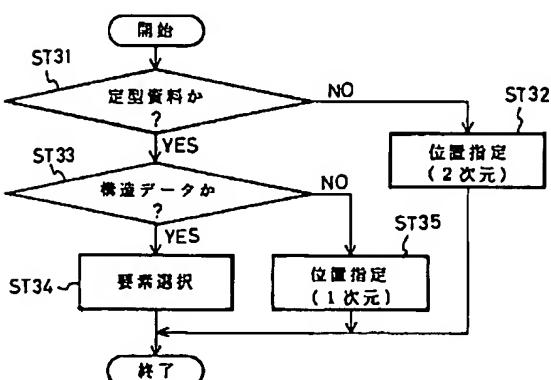
【図15】



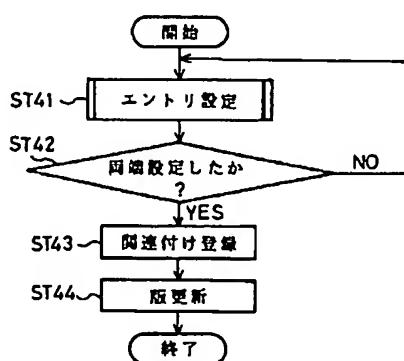
【図16】



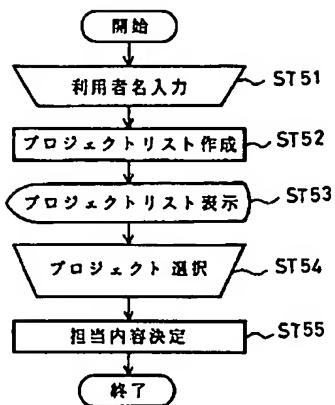
【図20】



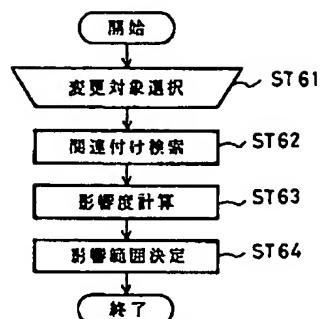
【図19】



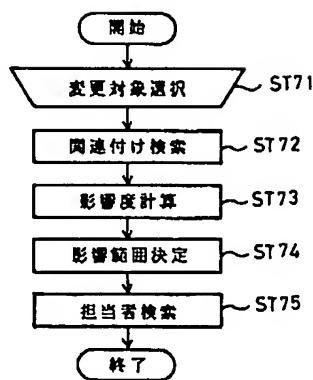
【図22】



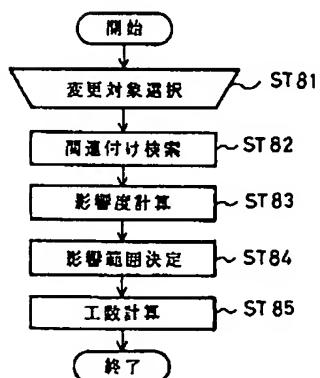
〔图23〕



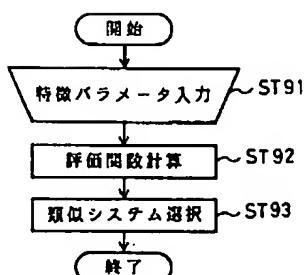
【図24】



【図25】



【図26】



〔图27〕

